AMENDMENTS TO THE CLAIMS

Claims 1-6, 8-13, 15-20, and 22-29 were originally pending.

Please amend claims 2-6, 9-13, and 16-19.

Please cancel claims 1, 8, 15, 20, and 22-29 without prejudice.

Accordingly, claims 2-6, 9-13, and 16-19 remain pending.

The following listing of claims replaces all prior versions, and listings of claims in the application.

1. (Canceled).

2. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 6~~1~~, wherein the dynamic variable amount of time is based on a maximum amount of time that a thread can yield before needing to be scheduled for execution.

3. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 6~~1~~, wherein the device is a battery powered device.

4. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 6~~1~~, wherein the operating system is a Microsoft WINDOWS CE, Linux, WindRiver, QNX, or PALM operating system.

5. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 6~~1~~, wherein the predetermined periodic rate is a millisecond.

6. (Currently amended) ~~A method as recited in claim 1:~~ A computer-implemented method for providing thread scheduling in a device, the device comprising one or more hardware elements operatively coupled to an operating system comprising a plurality of program modules, the method comprising:

scheduling one or more threads according to a predetermined periodic rate;

~~wherein the providing further comprises~~ setting a system timer to generate a notification at the predetermined periodic rate;

determining whether or not there are any threads to execute;

responsive to a determination that there are no threads to execute:

(a) deactivating one or more of the hardware elements and the program modules for a dynamic variable amount of time, the dynamic variable amount of time being independent of the predetermined periodic rate and being based on a sleep state of a set of threads in a sleep queue; and

(b) ~~wherein the deactivating further comprises~~ resetting the system timer to generate the notification after the dynamic variable amount of time has elapsed since the deactivating; and

wherein the method further comprises:

receiving the notification after the dynamic variable amount of time has elapsed since the deactivating; ~~and~~

responsive to the receiving:

resetting the system timer to generate the notification at the predetermined periodic rate; and

activating the one or more of the hardware ~~modules~~ elements and the program modules.

7-8. (Canceled).

9. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 13~~8~~, wherein the device comprises a battery powered device.

10. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 13~~8~~, wherein the operating system comprises a Microsoft WINDOWS CE operating system.

11. (Currently amended) ~~A~~ The method ~~as recited in~~ of claim 13~~8~~, wherein the predetermined periodic rate is a millisecond.

12. (Currently amended) ~~A method as recited in claim 8:~~ A method for providing thread scheduling in a device, the device comprising one or more hardware elements operatively coupled to an operating system comprising a plurality of program modules, the method comprising:

scheduling one or more threads at a predetermined periodic rate, ~~wherein~~ the scheduling ~~further comprises~~ comprising setting a system timer to the predetermined periodic rate, the predetermined periodic rate corresponding to a thread scheduling accuracy; ~~and~~

determining whether or not there are any threads to execute;

responsive to a determination that there are no threads to execute:

(a) deactivating one or more of the hardware elements and the program modules for a dynamic variable amount of time, the dynamic variable amount of time being based on a sleep state of a set of threads in a sleep queue and independent of the predetermined periodic rate; and

(b) ~~wherein the deactivating further comprises~~ resetting the system timer to generate a notification after the dynamic variable amount of time has elapsed since the deactivating; and

activating the one or more of the hardware elements and the program modules only when the operating system:

(c) needs to schedule a thread for execution upon expiration of the dynamic variable amount of time since the deactivating, or

(d) upon receipt of an external event that is not a system timer event.

13. (Currently amended) ~~A method as recited in claim 8:~~ A method for providing thread scheduling in a device, the device comprising one or more hardware elements operatively coupled to an operating system comprising a plurality of program modules, the method comprising:

scheduling one or more threads at a predetermined periodic rate;

determining whether or not there are any threads to execute;

responsive to a determination that there are no threads to execute:

(a) deactivating one or more of the hardware elements and the program modules for a dynamic variable amount of time, the dynamic variable amount of time being based on a sleep state of a set of threads in a sleep queue and independent of the predetermined periodic rate; and

(b) ~~wherein the deactivating further comprises~~ resetting a system timer to generate a notification after the dynamic variable amount of time has elapsed, the dynamic variable amount of time being a maximum amount of time that a thread can yield to other threads before needing to be scheduled for execution; ~~and~~

activating the one or more of the hardware elements and the program modules only when the operating system needs to perform an action selected from a group of actions comprising scheduling a thread for execution upon expiration of the dynamic variable amount of time since the deactivating, or upon receipt of an external event that is not a system timer event, ~~wherein~~ the activating ~~further comprises~~ comprising resetting the system timer to the predetermined periodic rate to provide substantial thread scheduling accuracy.

14-15. (Canceled).

16. (Currently amended) ~~A~~ The computer-readable storage medium ~~as recited in~~ of claim 19~~15~~, wherein the dynamic variable amount of time comprises a maximum amount of time that a thread has specified that it will yield to other threads before it needs to be scheduled for execution.

17. (Currently amended) ~~A~~ The computer-readable storage medium ~~as recited in~~ of claim 19~~15~~, wherein the device comprises a battery powered device.

18. (Currently amended) ~~A~~ The computer-readable storage medium ~~as recited in~~ of claim 19~~15~~, wherein the operating system comprises a Microsoft WINDOWS CE operating system.

19. (Currently amended) ~~A computer-readable storage medium as recited in claim 15, wherein the computer-executable instructions further comprise instructions for:~~ A computer-readable storage medium containing computer-executable instructions for scheduling threads in a device, the device including an operating system comprised of a plurality of program modules that are in turn coupled to one or more hardware elements, the computer-executable instructions comprising instructions for:

determining at a periodic rate whether or not there are any threads to execute; and

responsive to a determination that there are no threads to execute:

(a) deactivating one or more of the program modules and the hardware elements for a dynamic variable amount of time, the dynamic variable amount of time being independent of the periodic rate, the dynamic variable amount of time being based on a sleep state of a set of threads in a sleep queue; and

(b) ~~in the deactivating,~~ configuring a system timer to send a first timer interrupt after the dynamic variable amount of time has elapsed, the dynamic variable amount of time being a maximum amount of time that a first thread can yield to a second thread before the first thread needs to be executed;

responsive to receiving the first timer interrupt:

(~~a~~c) configuring the system timer to send a second timer interrupt at the periodic rate; and

(~~b~~d) activating the one or more of the program modules and the hardware elements to determine if there are any threads to execute.

20-29. (Canceled).